［研究ノート］

低域部の周波数解析精度の改善とオーディオ-MIDI 変換ツール開発への応用

Improvement of Frequency-Analysis Precision for Low-frequency Components and Application to Development of Audio-MIDI Conversion Tool

茂出木 敏雄
Toshio Modegi

［抄録］
MIDI カラオケなどの自動演奏データを作成する用途や、耳コピーの自動化や自動採譜の用途において、オーディオ信号を MIDI 形式に変換するツールの実現が要望される。近年、生成 AI 技術が進歩しており、生成される音楽作品の品質を向上させるため、膨大な音楽作品により訓練する必要がある。既製のオーディオデータを変換して、機械学習向けの MIDI データを提供する用途も考えられる。いずれの場合も、与えられたオーディオ信号を MIDI 形式に変換する信号処理系の精度が要求され、それを左右するのが時間周波数解析である。しかし、低域部の周波数解析には次の２つの問題がある。低域部を忠実に解析しようとすると時間分解能が低下すること。もう１つは、低域の基音成分については音響信号に記録されず、音響解析では対応できない場合があること。本稿では、一般化調和解析手法に改良を加え、低域部の周波数解析精度を向上させ、欠落している基音成分を補間する機能を搭載させた、オーディオ-MIDI 変換ツールを開発したので、その結果を報告する。

Abstract:
Realization of Audio-MIDI conversion tools are requested for creation of playback-control data such as for MIDI-Karaoke machines, and for development of an automatic music notation system. In these days, a Generative-AI technique is highly advanced. For increasing quality of music generation, we can also provide MIDI contents for machine learning by converting a large quantity of waveform-audio contents. In these applications, precisions of converted MIDI data from audio signals and time-frequency analysis play an important role. However, there are two kinds of problems on analysis of low-frequency components. One is difficult to increase resolution of frequency analysis without sacrificing temporal analysis resolution. The other is difficult to analyze missing fundamental components in recorded audio signals. In this paper, we propose an improved generalized harmonic analysis method, which can increase precision of low-frequency components and compensate predicted fundamental components in a time-frequency analysis. And we report applying our improved analysis method, to our developing audio-MIDI conversion tool.

キーワード：自動採譜, オーディオ-MIDI 変換, 低域部, 一般化調和解析, ミッシング・ファンダメンタル, 生成 AI
Keywords: automatic music notation, audio-MIDI conversion, low-frequency component, generalized harmonic analysis, missing fundamental, generative AI

１．はじめに

MIDI カラオケ、着信メロディーや BGM 再生機など向けに MIDI 形式の演奏データを作成する用途や、録音楽曲より耳コピーを行って譜面を起こす、耳コピー支援や自動採譜の用途において、オーディオ信号を MIDI 形式に変換するツールの実現が要望される [1)]。前者においては、録音楽曲より MIDI のベロシティ（演奏音の強弱）やピッチベンド（ビブラートなど半音未満の微小な音高の揺れ）に対応する表情パラメータ [2)]を認識して生演奏に近い演奏表現が求められる。これに対し、後者においては、録音楽曲に含まれる倍音や、テンポ、強弱、ピッチ等の演奏表現上のゆらぎを除去して元の譜面に近い音符の表現が求められる。

また、近年では生成 AI 技術が急速に進歩しており、文章・画像・音楽など種々のメディアのコンテンツをプロンプトと呼ばれる文字列による指示に基づき自動生成可能なツールが登場している。この技術において生成される作品の品質を左右するのが、事前の機械学習に使用される膨大な既製のコンテンツである。OpenAI 社の「MuseNet[7)]」や Google 社の「MusicLM[8)]」などの音楽生成 AI では、機械学習させるための大量の MIDI 形式や波形形式の音源データが必要となる。特に「MuseNet[7)]」で必要とする MIDI データは基本的に打ち込み等により手動で制作されているため、録音により取得可能な波形形式のサンプリング音源に比べ桁違いにコンテンツが少ない。

そこで、第３の用途として、音楽生成 AI 向けに、波形形式の音源データから MIDI 形式に自動変換して、機械学習向けに提供する MIDI 形式の音源データを拡充するアイデアが考えられる。音楽生成 AI の仕様により、前者の自動演奏用途に対応する演奏表現を伴う MIDI データが必要になる場合や、後者の自動採譜用途に対応する譜面に近い MIDI データが必要になる場合の双方が考えられる。

前述のいずれの用途においても、与えられたオーディオ信号を MIDI 形式に変換する信号処理ツールの開発が要望される。前者に対しては、「オート符」[2)-4)]が、後者に対しては、「採譜の達人」[5)]や「WaveTone」[6)]などオーディオ信号を MIDI 形式および五線譜に変換するフリーウェアのツールが幾つか開発されているが、いずれも業務に対応できる性能には至っていない。

その主な理由として、オーディオ信号を MIDI 形式に変換する信号処理のコアである時間周波数解析において、文献 14)で述べたように、量子力学の不確定性原理として知られる物理学的な限界があるためである。即ち、周波数分解能と時間分解能にはトレードオフの関係があり、双方の次元を同時に高精度に解析できないという問題がある。これに加え、MIDI 形式の音楽音階に変換する際の特有の問題として、次の２点がある。

第１の問題は、解析する周波数の次元はリニアではなく、ヒトの知覚特性に基づいて、音楽音階がノートナンバーと呼ばれる対数スケールになっている点である。ノートナンバーが小さい低域部では、周波数間隔が狭いため、高域部に比べ細かい周波数分解能が要求される。前述の不確定性原理の制約の基で、時間分解能をできるだけ維持しながら、周波数分解能を向上させる工夫が必要である。

第２の問題は、ミッシング・ファンダメンタル [10)]と呼ばれ、採譜対象の低域音が録音信号に収音されず欠落する場合がある点である。低域部のボーカルの基本周波数(F0)や楽音の基音は、発声器官や楽器などの音源機器の周波数特性や、録音時の音響機器の周波数特

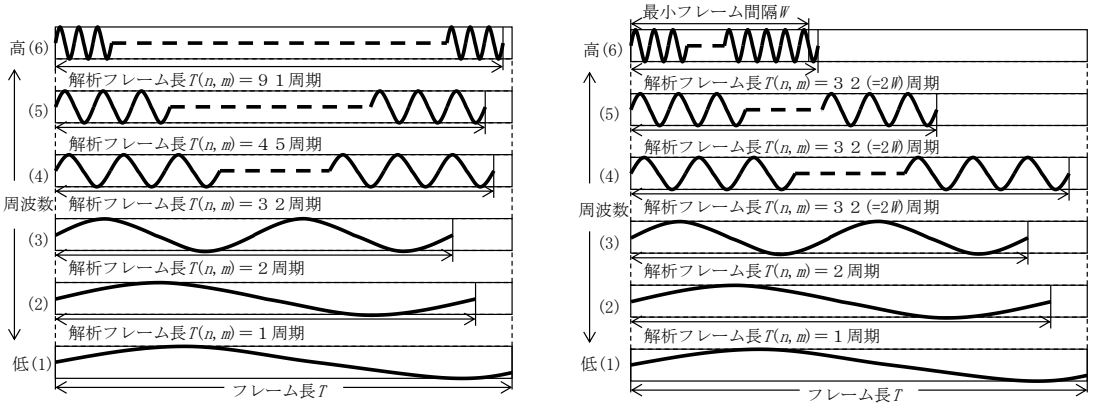
性により収音されない場合がある。しかし、ヒトは、録音信号に高次のフォルマントや倍音成分が収音されていれば、それらの高次成分を基に、欠落している基音を認識することができる。そのため特に、自動採譜の用途においては、収音されている倍音ではなく、欠落している基音を採譜対象として推定する必要が生じる場合がある。

本稿では文献 14)と同様に、周波数解析として一般化調和解析(GHA, Generalized Harmonic Analysis)[11]を採用し、その後の処理に前述の２つの問題に対応する改良を加えた。前述の第１の問題に対応するため、周波数解析後の解析音素の連結処理において、ノートナンバーごとに連結条件を可変にするような改良を加え、周波数分解能を向上させても、時間分解能の劣化を抑えるようにした。また、第２の問題に対応するため、周波数解析後の倍音成分補正処理に改良を加え、倍音を抑圧しながら、最大 8 倍音までの倍音成分を基に基音を推定して追加する機能を搭載した。

また、既提案の一般化調和解析 [11]は、短時間フーリエ変換 [9]を繰り返し実行して、解析スペクトルの各強度を順次決定する方法である。短時間フーリエ変換を実行してピーク周波数の強度を１つ決定したら、次の短時間フーリエ変換を実行する前に、原音信号からピーク周波数をもつ調和関数成分を減算させるようにしている。この調和関数成分との差分を算出する過程で重畳する疑似信号成分を抑圧する手法を、文献 14)では併せて提案したが、その後の研究により逆効果であることが判明し、本稿では先提案を不採用とした。

本稿では、これらの改良を加えた信号処理系を、既開発のオーディオ-MIDI 変換ツールに実装し、ピアノ 88 鍵に対応する正弦波やピアノ・サンプリング音源を用いて、前述の問題に対する改善効果を報告する。併せて、文献 14)に対する比較評価結果についても報告する。

２．既提案の一般化調和解析に使用する解析フレームの設定方法



(A)短時間フーリエ変換に使用される調和関数　　(B)既提案の一般化調和解析に使用される調和関数

図１　既提案の短時間フーリエ変換と一般化調和解析に使用される調和関数

表１　ノートナンバーに対する微分音分解能の設定表

ノートナンバーn	0	・	62												
微分音分解能$M(n)$	1	・	1												
ノートナンバーn	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
微分音分解能$M(n)$	3	3	3	3	3	3	3	3	5	5	5	5	5	5	7
ノートナンバーn	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92
微分音分解能$M(n)$	7	7	7	9	9	9	9	11	11	11	13	13	15	15	17
ノートナンバーn	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107
微分音分解能$M(n)$	17	19	19	21	23	25	27	29	31	33	35	37	39	41	43
ノートナンバーn	108	109	・	127											
微分音分解能$M(n)$	45	45	・	45											

図１-(A)は既提案の短時間フーリエ解析法で使用する調和関数を示す。調和関数としては正弦波および余弦波を使用し、周波数としては基本的に MIDI の半音階のノートナンバーn（0≤n≤127）に対応する 128 種の周波数$f(n)$を以下のように与える。

$$f(n) = 440 \cdot 2^{(n-69)/12} \quad [Hz] \quad (1)$$

ただし、周波数の分解能として半音間隔では不十分で、更に文献 2)にあるような微分音・ピッチベンドのパラメータを解析するには、より細かい間隔で解析が求められる。といっても、最小のセント間隔で 12800 種の周波数の調和関数を使用して短時間フーリエ解析を行うのも現実的ではない。(1)式により、ノートナンバーn の周波数 $f(n)$はノートナンバーが大きくなるほど指数関数的に大きくなるので、ノートナンバー間の周波数の間隔もノートナンバーが大きくなるほど大きくなる。そこで、表１に示すように、ノートナンバーn が大きくなるにつれ、隣接ノートナンバー間（半音）で解析周波数の間隔があまり広がらないように、指数関数的に大きな値をもつ $M(n)$種に分割した周波数をもつ調和関数を準備して解析を行う（ただし、処理負荷軽減のため実用上は、n>98 では $M(n)$=25 に設定）。即ち、分割した周波数をもつ微分音を m（0≤m≤$M(n)$–1）として、ノートナンバーn（0≤n≤127）では、以下式で示される $M(n)$種の周波数 $f(n,m)$をもつ調和関数を用いて解析を行う。

$$f(n,m) = 440 \cdot 2^{(n-69+\frac{m}{M(n)})/12} \quad [Hz] \quad (2)$$

短時間フーリエ解析法では、解析窓のフレーム長を T とすると、与えられたサンプリング周波数 Fs [Hz]の音響信号より区間 T サンプルだけ切り出して各調和関数と相関計算を行う。その際、相関計算を行う範囲である解析フレーム長 $T(n,m)$ [単位:サンプル]は、$T(n,m)$≤T の条件で、以下のように周期 $Fs/f(n,m)$の整数倍で最大になる Cy 周期分に設定する。

表２　ノートナンバーに対する解析フレーム長の設定表

ノートナンバー n	(1)フレーム長:2048		(2)フレーム長:4096		(3)フレーム長:8192	
	周期 Cy	解析フレーム長 $T(n,m)$	周期 Cy	解析フレーム長 $T(n,m)$	周期 Cy	解析フレーム長 $T(n,m)$
21	1	1603（36msec）	2	3207（72msec）	5	8017(181msec)
29	2	2020	4	4040	8	8001
36	3	2022	6	4045	12	8090
41	4	2020	8	4040	16	8081
45	5	2004	10	4009	20	8018
48	6	2022	12	4045	24	8090
51	7	1984	14	3968	28	7937
53	8	2020	16	4040	32	8081(183msec)
66	16	1944	32	3888（88msec）	32	3888
78	32	1954（44msec）	32	1954	32	1954
108	32	346(7.8msec)	32	346(7.8msec)	32	346(7.8msec)

$$T(n,m) = MAX\left[\frac{Cy \cdot Fs}{f(n,m)}\right] \qquad (3)$$

図１-(A)の例では、いずれの周波数においても、解析フレーム長 $T(n,m)$は T に近い値となり、周波数分解能は解析対象のノートナンバーn とともに向上するが、時間分解能は固定である。しかし、図１-(A)の例では、最下位(1)の周波数ではフレーム長 T に１周期分の調和関数を収納できていないため、この周波数では精度の良い解析は行えない。フレーム長 T を大きく設定すれば、最下位(1)の周波数を含め周波数解析精度は向上するが、時間分解能が全体的に低下してしまう。逆に、フレーム長 T を小さく設定すれば、時間分解能は向上するが、最下位(1)の周波数を含め低い周波数では精度の良い解析は行えなくなる。即ち、時間と周波数の解析精度はトレードオフの関係になり、文献 14)で述べたように、時間と周波数に関する不確定性原理とよばれる。

本稿では、文献 14)で提案した、図１-(A)の短時間フーリエ変換を基本として、図１-(B)のように全域にわたって周波数解析精度を維持しながら、中域から高域における時間分解能を向上させる方法を採用する。図１-(B)の周波数(1)～(4)まで、具体的には後述する周波数解析を行う際の最小フレーム間隔を W（Fs=44100 の場合 W=16）として、周期が $Cy \leq 2W$ までの低域周波数では図１-(A)と同様に(3)式に基づいて解析フレーム長 $T(n,m)$を設定する。$Cy>2W$ となる中域周波数では、$Cy=2W$ に固定して解析フレーム長 $T(n,m)$を $2W$ 周期分 $2W \cdot Fs / f(n,m)$ とする可変長窓に設定する。これにより、周波数解析精度をあまり低下させずに時間分解能を向上させることができる。ただし、高域周波数では $Cy=2W$ のままでは周波数解析精度が低下してしまう。そのため、解析フレーム長 $T(n,m)<W$ の場合、$T(n,m) \geq W$ の

条件で、以下のように周期 $Fs/f(n,m)$の整数倍で最小になる Cy 周期分に設定する。

$$T(n,m) = MIN\left[\frac{Cy \cdot Fs}{f(n,m)}\right] \qquad (4)$$

フレーム長 T を、(1) T=2048, (2) T=4096, (3) T=8192 に設定した場合、主要なノートナンバーn における周期 Cy と解析フレーム長 $T(n,m)$の具体例を、表２に示す。サンプリング周波数が 44.1[kHz]で、W=16 に設定している場合、ノートナンバーn<53 となる低域部では、Cy<32 の範囲まで Cy の値はノートナンバーn に伴なって大きくなるが、解析フレーム長 $T(n,m)$はフレーム長 T に近い値にほぼ一定となる。ノートナンバーn≥53 となる中・高域部では、$T(n,m)$≥W の条件で、周期 Cy が 32 に一定になり、解析フレーム長 $T(n,m)$はノートナンバーn に伴なって小さくなる。ノートナンバーn≥78 となる高域部では、フレーム長 T が(1)(2)(3)のいずれの場合でも、解析フレーム長 $T(n,m)$は同一の値になり、Cy>32 になることはなかった。具体的な時間分解能は、低域部では(1) 46msec, (2) 92msec, (3) 185msec となり、中域部より各々徐々に小さくなり、高域部ではいずれも最小の 7.8msec となる。

３．提案する時間周波数解析法を用いたオーディオ-MIDI 変換ツールの概要

本稿で提案するオーディオ-MIDI 変換ツールの処理構成は、文献 12)に基づいている。はじめに、与えられたソース音響信号より周波数解析対象の解析フレームを抽出するが、後続する解析フレームとのフレーム間隔はソース音響信号の周波数変動を大まかに検出しながら適応的に可変設定するようにしている。即ち、周波数成分の変化が大きい箇所ではフレーム間隔を狭くし、周波数成分の変化が小さい箇所ではフレーム間隔を大きくする。続いて、抽出した解析フレームに対して、前章で述べた一般化調和解析に基づいて周波数解析を行う。このとき、ノートナンバーに対応する周波数ごとに隣接するノートナンバーとの半音間を、表１に示した微分音分解能に基づいて微分音（副周波数）に分割して解析を行う。最後に、時間的に隣接して近傍の主周波数をもつ解析成分（解析音素）を連結し音符としてまとめ、MIDI イベント形式で符号化する。具体的には、算出される音長（デュレーション）だけ時刻がずれた２つのノートオンとノートオフのイベントで符号化される。また、微分音解析の結果を基に、ノートオンとノートオフのイベントの間にピッチベンドやエクスプレッションなどの表情制御コードを符号化して挿入することもできる。

図２は、文献 12)-14)で提案されているオーディオ-MIDI 変換処理に、本稿提案の機能を追加した、具体的な処理構成を示す。文献 13)では、処理(B-1)から処理(B-3)に示される可変長フレーム間隔の時間周波数解析の処理構成を提案し、処理(B-3)に一般化調和解析に基づいた周波数解析手法を組み込んでいる。本稿では、この後に、オプションで後述する基音推定・追加を行う処理(C)を追加している。続いて、オプションで倍音除去を実現する処理(D)の倍音成分補正と、処理(F)の解析音素の連結処理では、既提案の方法に本稿独自の改良を加えている。処理(F)から処理(H)までは既提案と同様な構成で、処理(F)のノートイベントの符号化処理においてオプションでピッチベンド符号化を実現できるようにしている。以下、処理(A)から処理(H)の各々に対して、８つの節に分けて説明する。

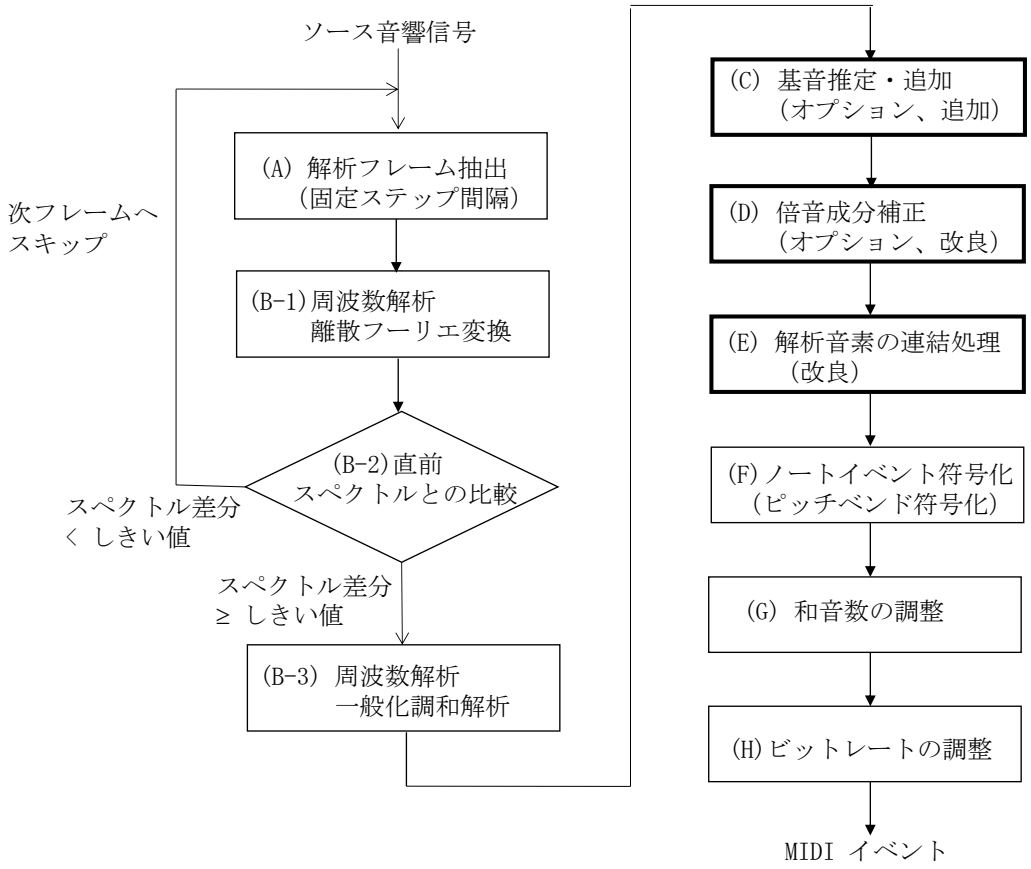


図２　提案するオーディオ-MIDI 変換ツールの具体的な処理構成

３．１．　解析フレーム抽出（固定ステップ間隔）(A)

時間周波数解析では解析フレームを時間軸方向に移動させながら、信号全体の解析を行うが、この際のフレーム長 T とフレーム間隔の設定方法について以下述べる。

周波数分解能はフレーム長により変化し、経験上ソース音響信号のサンプリング周波数 Fs が 44.1[kHz]の場合、ピアノ鍵盤の最低音（ノートナンバー:21）まで忠実に解析するためには、フレーム長 T として表２に示されるように 2048 サンプル以上必要である。解析計算範囲の解析フレーム長 $T(n,m)$はフレーム長 T の範囲内で解析周波数ごとに可変に設定するが、フレーム長 T は上限値として、例えば 4096 を与える。そうすると、ノートナンバー21 の場合、$Fs/f(n,m)$=1603 サンプルとなり、Cy=2[cycle]となる。ただ、その後の実験により、Cy=2[cycle]では低域部において十分な解析精度が得られず、安定した解析には、Cy=4[cycle]、フレーム長 T として 8192 サンプル必要であることが判明した。

一方フレーム間隔は、小さくするほど時間分解能が向上するが計算時間も増大する（ただし、併せてフレーム長も小さくしないと顕著な時間分解能の向上効果はない)。そして、解析対象信号が単調である箇所に対して、必要以上にフレーム間隔を細かくすると、後述する解析音素の連結処理で支障をきたす。そこで、効率的な計算および高精度な解析音素

の連結処理のためにも、フレーム間隔は解析対象フレームごとに変化させ、適応的に設定する方法が望ましい。本稿では文献 13)の提案に基づき、周波数解析時において一般化調和解析による高精度な周波数解析を行う前に、フレーム間隔の最小値である固定値の最小フレーム間隔 W（例えば、T=4096 の場合 W=16 サンプル）で離散フーリエ変換を行い、周波数変化が顕著な箇所を、高精度な周波数解析を行う箇所として探索する方法をとる。

３．２．　周波数解析・一般化調和解析(B)

３．２．１．　周波数解析・離散フーリエ変換(B-1)

前節で述べた方法により、サンプリング周波数 F_s の原音響信号より p 番目に抽出された解析フレームのサンプル配列を $x(p,i)$ $(0 \le i \le T-1)$とする。本周波数解析は、ノートナンバー n $(0 \le n \le 127)$に対して、表１に基づいて $M(n)$個の微分音 m を定義し、(2)式に基づく 128×$M(n)$種の解析周波数 $f(n,m)$の調和関数を用いて短時間離散フーリエ変換により行う。この微分音を用いた解析は周波数解析精度を向上させることが主目的であるが、後述するオプション処理により、この半音未満の精度で微分音解析された結果を、ピッチベンドなどの表情制御コードへの符号化に使用することもできる。

p 番目の解析フレーム $x(p,i)$に対して、ノートナンバー分の短時間フーリエ変換による相関配列 $E(p,n)$ $(0 \le n \le 127)$と副周波数配列 $S_o(p,n)$および $S(p,n)$を定義し、$0 \le n \le 127$ および $0 \le m \le M(n)-1$ に対して以下式で相関計算を行う。副周波数配列 $S_o(p,n)$および $S(p,n)$には、相関が合った微分音 m の値と、m を最大値 M_o（例えば、M_o=25）の微分音分解能に換算した値 m'（$m'=m \cdot M_o / M(n)$, $0 \le m' \le M_o-1$）を各々収納する。(5)式において $T(n,m)$は解析フレーム長で、前章で述べた通りフレーム長 T を超えない範囲で周波数ごと可変に設定する。

$$A(p,n,m) = \frac{1}{T(n,m)} \sum_{i=0}^{T(n,m)-1} \left(x(p,i) \cdot \sin\left(\frac{2\pi f(n,m) \cdot (i+pW)}{F_s} \right) \right)$$

$$B(p,n,m) = \frac{1}{T(n,m)} \sum_{i=0}^{T(n,m)-1} \left(x(p,i) \cdot \cos\left(\frac{2\pi f(n,m) \cdot (i+pW)}{F_s} \right) \right)$$

$$C(p,n,m) = A(p,n,m)^2 + B(p,n,m)^2 \qquad (5)$$

ここで、p>0 で $A(p-1,n,m)$と $B(p-1,n,m)$の値が既知の場合、(5)式の $A(p,n,m)$と $B(p,n,m)$を算出する式は(5')式のように変形でき、直前解析フレーム $p-1$ における相関計算結果を用いて、計算範囲を縮小でき高速に算出できる。

$$A(p,n,m) = A(p-1,n,m) - \frac{1}{T(n,m)} \sum_{i=0}^{W-1} \left(x(p-1,i) \cdot \sin\left(\frac{2\pi f(n,m) \cdot (i+(p-1)W)}{F_s} \right) \right)$$
$$+ \frac{1}{T(n,m)} \sum_{i=T(n,m)-W}^{T(n,m)-1} \left(x(p,i) \cdot \sin\left(\frac{2\pi f(n,m) \cdot (i+pW)}{F_s} \right) \right)$$

$$B(p,n,m) = B(p-1,n,m) - \frac{1}{T(n,m)} \sum_{i=0}^{W-1} \left(x(p-1,i) \cdot \cos\left(\frac{2\pi f(n,m) \cdot (i + (p-1)W)}{F_s} \right) \right)$$

$$+ \frac{1}{T(n,m)} \sum_{i=T(n,m)-W}^{T(n,m)-1} \left(x(p,i) \cdot \cos\left(\frac{2\pi f(n,m) \cdot (i + pW)}{F_s} \right) \right)$$

$$C(p,n,m) = A(p,n,m)^2 + B(p,n,m)^2 \qquad (5')$$

続いて、ノートナンバーn ごとに、0≤m≤$M(n)$–1 の範囲で相関配列 $C(p,n,m)$を最大にする$C(p,n,m_{max})$を求め、$E(p,n)$=$C(p,n,m_{max})$、$S_o(p,n)$=m_{max}、$S(p,n)$=m_{max}·M_o/$M(n)$と算出する。

３．２．２．　直前スペクトルとの比較(B-2)

（5）式または（5’）式でノートナンバーn (0≤n≤N–1, N=128)ごとに算出された相関配列$E(p,n)$と直前解析フレームにおける $E(p-1,n)$との差分割合の平均値 $dE(p-1,p)$を以下のように算出し、$dE(p-1,p)$が所定のしきい値（例えば 0.15、ピッチベンド符号化を行う場合は 0.2）未満であれば、3.1 節に戻り次の解析フレーム抽出に進み、所定のしきい値以上であれば、次の 3.2.3 節の周波数解析・一般化調和解析へ進む。

$$dE(p-1,p) = \frac{1}{N} \sum_{n=0}^{N-1} \left\{ \frac{|E(p,n) - E(p-1,n)|}{E(p,n) + E(p-1,n)} \right\} \qquad (6)$$

３．２．３．　周波数解析・一般化調和解析(B-3)

解析フレーム p は q 番目に一般化調和解析を行う可変解析フレームであるとし、解析フレーム ID 配列を $P(q)$とすると、$P(q)$=p と設定し、可変解析フレーム q において、一般化調和解析による強度値 $E_o(q,n)$ (0≤n≤127)を定義し、初期値を全て–1 とする。

（ａ）ノートナンバーn に対して $E_o(q,n)$<0 で、かつ $E(p,n)$が最大になる $E(p,n_{max})$を求め、m_{max}=$S_o(p,n_{max})$とする。ただし、p=$P(q)$とする。(5)式を簡素化した以下(7)式を用いて$A(p,n_{max},m_{max})$および $B(p,n_{max},m_{max})$を再計算する。

$$A(p,n_{max},m_{max}) = \frac{1}{T(n_{max},m_{max})} \sum_{i=0}^{T(n_{max},m_{max})-1} \left(x(p,i) \cdot \sin\left(\frac{2\pi f(n_{max},m_{max}) \cdot i}{F_s} \right) \right)$$

$$B(p,n_{max},m_{max}) = \frac{1}{T(n_{max},m_{max})} \sum_{i=0}^{T(n_{max},m_{max})-1} \left(x(p,i) \cdot \cos\left(\frac{2\pi f(n_{max},m_{max}) \cdot i}{F_s} \right) \right)$$

$$E_o(q,n_{max}) = A(p,n_{max},m_{max})^2 + B(p,n_{max},m_{max})^2 \qquad (7)$$

（ｂ）上記決定した $A(p,n_{max},m_{max})$および $B(p,n_{max},m_{max})$を用いて、以下(8)式でサンプル配列 $x(p,i)$の全ての要素$(0\leq i\leq T-1)$を更新する。このとき、調和関数の成分が疑似的に重畳され、疑似ピークが生じることを防止するため、文献 14)では、p 番目に抽出された解析フレーム $x(p,i)$ $(0\leq i\leq T\text{-}1)$に対して振幅包絡線を算出し、解析フレーム内における振幅分布を重み関数 $w(p,i)$（$0\leq w(p,i)\leq 1$）として求め、調和関数に重み付けを行っていた。しかし、振幅包絡線に基づく歪みが加わり、疑似的な倍音が新たに発生することが発覚したため、本稿では(8)式のように調和関数への重み付けを行わないようにした。

$$x(p,i)=x(p,i)-A(p,n_{max},m_{max})\cdot\sin\left(\frac{2\pi f(n_{max},m_{max})\cdot i}{F_s}\right)$$
$$-B(p,n_{max},m_{max})\cdot\cos\left(\frac{2\pi f(n_{max},m_{max})\cdot i}{F_s}\right)\qquad(8)$$

(8)式に基づいてサンプル配列 $x(p,i)$を更新後、再度（ａ）の処理に戻り、$0\leq n\leq 127$ の全ての強度値 $E_o(q,n)$の値が 0 以上の値に決定されるまで（ａ）と（ｂ）の処理を繰り返す。

３．３．　基音推定・追加(C)

本稿では新規にオプションで基音推定・追加を行う処理を追加する。ここで、ミッシング・ファンダメンタル（基本の欠落）と呼ばれる聴覚現象[10]について説明する。図３-(A)(C)

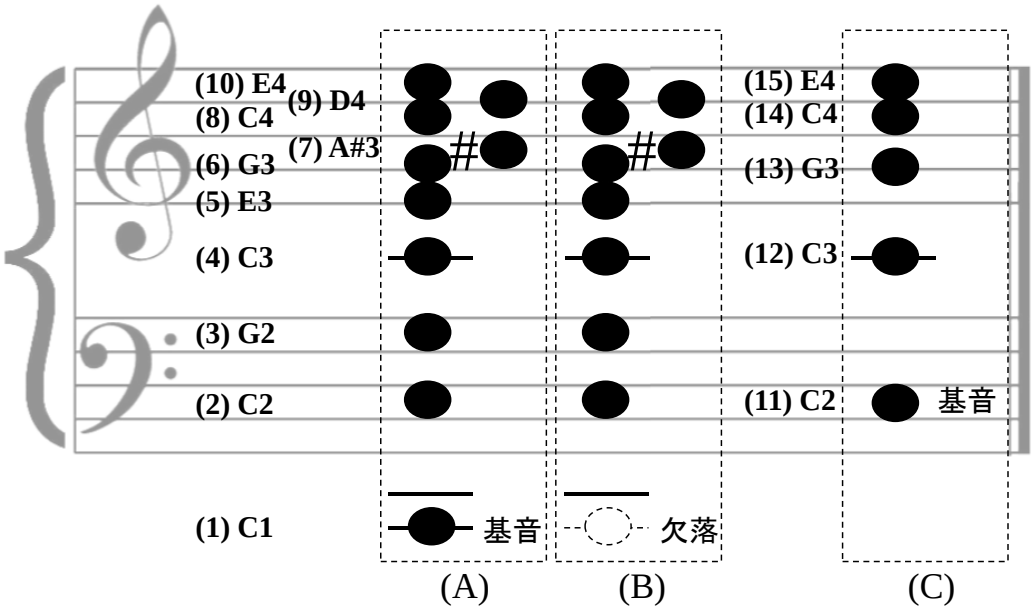


図３　ミッシング・ファンダメンタルの説明図

は、鍵盤で各々C1 と C2 の音をピアノで弾いたときに聴取される倍音系列を示す。これは C1 と C2 という 65 と 130 [Hz]を基本周波数とする整数倍の音列である。この時、楽器・MIDI 音源や収音機器の周波数特性により、図３-(B)のように図３-(A)の基音 C1 が物理的に減衰または欠落することがある。この時、図３-(B)の音列の最低音 C2 は、図３-(C)の音列の最低音である基音 C2 と音高は同一であるが、図３-(B)において聴覚的に認識される音高（ピッチ）は、図３-(C)の C2 のように聞こえず、図３-(A)の C1 のように聞こえる。

即ち、図３-(B)の音列に、図３-(A)の基音 C1 が仮想的に存在するように知覚され、図３-(B)の(2)から(10)の倍音系列を基に、基音 C1 を補完する機能がヒトの聴覚系にて働く。これが、ミッシング・ファンダメンタルである。楽器音やボーカルにおいて採譜や MIDI 変換の対象とするのは主として基本周波数（基音）の成分であるため、図３-(B)のような音列をもつ楽器音を採譜する場合、解析された音列の最低音 C2 ではなく、音列を基に推定される、最低音より低い基音 C1 を採譜する必要がある。

C2 音が基音であるか倍音であるかは、図３-(B)と図３-(C)の倍音系列の相違から判断できる。具体的には、C2 音が基音である場合、図３-(C)のように図３-(B)の G2 音や E3 音が存在しない。本稿では、以下のように、判断対象とする低域部の音に対して、音列上の当該音より高次の 4 個の倍音の分布を基に、基音か 2～8 倍音のいずれかを判断し、基音を補完する手法を提案する。

ノートナンバーn で、低域部 12≤n≤63 における強度値 $E_o(q,n)$を、n=12 から n=63 の順に次の通り補正する。$E_o(q,n)$を基音、2 倍音～8 倍音のいずれかであると仮定して、$E_o(q,n)$に対する 4 つ倍音成分の総和 $S_{um}(b)$($0\le b\le 7$)を以下(9)式により 8 通り算出し、$S_{um}(b)$が最も大きくなる b_{max} を求める。この時、総和演算を行う倍音成分は短時間フーリエ変換による相関配列 $E(p,n)$の値を使用する。基音、2 倍音～8 倍音の周波数に対応する 8 個のノートナンバー・オフセットテーブル $N_o(b)$ (b=0,…,7)を、$N_o(b)$={0, 12, 19, 24, 28, 31, 34, 36}と定義して、$n\ge N_o(b_{max})$の場合、$E_o(q,n-N_o(b_{max}))$の値に、$E_o(q,n)$を加算する補正を行う。

$$
\begin{aligned}
S_{um}(0) &= E(p,n+12)+E(p,n+19)+E(p,n+24)+E(p,n+28)\\
S_{um}(1) &= E(p,n+7)+E(p,n+12)+E(p,n+16)+E(p,n+19)\\
S_{um}(2) &= E(p,n+5)+E(p,n+9)+E(p,n+12)+E(p,n+15)\\
S_{um}(3) &= E(p,n+4)+E(p,n+7)+E(p,n+10)+E(p,n+12)\\
S_{um}(4) &= E(p,n+3)+E(p,n+6)+E(p,n+8)+E(p,n+10)\\
S_{um}(5) &= E(p,n+3)+E(p,n+5)+E(p,n+7)+E(p,n+9)\\
S_{um}(6) &= E(p,n+2)+E(p,n+4)+E(p,n+6)+E(p,n+8)\\
S_{um}(7) &= E(p,n+2)+E(p,n+4)+E(p,n+5)+E(p,n+7) \qquad (9)
\end{aligned}
$$

３．４．　倍音成分補正(D)

文献 14)では、上記算出された $0\le n\le 127$ の全ての強度値 $E_o(q,n)$に対して、低音部から順に対象音を 2～10 倍音と仮定して、各々の基音に対応する強度値を基に減衰補正を行っていた。この方法では、減衰強度γをいくら強くしても残留倍音が生じる問題があった。そこで、本稿では、上記算出された $0\le n\le 127$ の全ての強度値 $E_o(q,n)$に対して、逆に高音部から順に対象音を基音と仮定して、2～10 倍音に対応する強度値に対して減衰補正を行うよう

にした。

上記算出された 0≤n≤127 の全ての強度値 $E_o(q,n)$に対して、2, 3, 4, 5, 6, 7, 8, 9, 10 倍の周波数に対応する 9 要素のノートナンバー・オフセットテーブル $N_o(b)$ (b=0,…,8)を、$N_o(b)$={ 12, 19, 24, 28, 31, 34, 36, 38, 40}、と定義して、次の通り補正を行う。そして、n=115 から n=0 の順に強度値 $E_o(q,n)$を更新し、0≤n+$N_o(b)$≤127 の場合、ノートナンバーn+$N_o(b)$に対応する強度値 $E_o(q,n+N_o(b))$を次式の通り補正する。この時、倍音の次数 b の値に反比例して補正割合を減衰させる。

$$E_o'(q,n+N_o(b)) = E_o(q,n+N_o(b)) - \frac{2\gamma}{(b+2)}\sqrt{E_o(q,n)E_o(q,n+N_o(b))} \qquad (10)$$

γは、正の実数値で倍音補正強度を与える。通常の楽曲ではγ>0.2 に設定して倍音補正を行う。ボーカルを含む音響信号で、フォルマント成分を倍音として残しておく必要がある場合は、γ=0 に設定し倍音補正を行わないようにする。式(10)による補正後の E_o'(q,n)が負値の場合、E_o'(q,n)=0 とする。(10)式により補正された強度値 E_o'(q,n)を $E_o(q,n)$として次ステップの解析音素の連結処理(E)以降は補正後の値を適用する。

３．５．　解析音素の連結処理(E)

q 番目の可変解析フレームにより周波数解析されたノートナンバーn の解析音素の成分を［時刻 $Time(q)$，時間長 $Length(q)$，主周波数 n，副周波数 $S(P(q),n)$，強度 $E_o(q,n)$］とし、前方に位置する解析音素との連結可能性パラメータ $Conn$（初期値、$Conn$=0）を設定する。はじめに、直前に隣接する q−1 番目の可変解析フレームが存在する場合、q−1 番目の可変解析フレームにより周波数解析されたノートナンバーn の解析音素の成分を［時刻 $Time(q-1)$, 時間長 $Length(q-1)$, 主周波数 n, 副周波数 $S(P(q-1),n)$, 強度 $E_o(q-1,n)$］とする。q−1 番目の解析音素が存在しない場合、$Conn$=0 とする。時刻 $Time(q)$および $Time(q-1)$は各々$P(q)$番目および $P(q-1)$番目の解析フレームの第１サンプルの原音響信号上の絶対サンプルアドレスをサンプリング周波数 Fs で除算することで得られる。時間長 $Length(q)$は $\{Time(q+1)-Time(q)\}\cdot\delta$で、時間長 $Length(q-1)$は $\{Time(q)-Time(q-1)\}\cdot\delta$で与えられる。隣接する可変解析フレームの時刻の差をそのまま時間長に設定すると音の切れが悪くなるため、１より小さい係数δ（例えば、δ=0.77）を乗算する。また、q+1 番目の可変解析フレームが存在しない場合、時間長 $Length(q)$は $T\cdot\delta$とする。

互いに時間的に隣接する q−1 番目および q 番目の解析フレームの２つ解析音素に対して、ノートナンバーn において上下±1 の変移を考慮し、副周波数を考慮した、q−1 番目と q 番目の解析フレーム間の周波数の差が所定値 N_{dif} 未満で、双方の強度が各々所定のしきい値 L_{min} 以上でかつ双方の強度の差が所定値 L_{dif} 以下で両者の連続性が認められる場合、即ち、以下(11-1)～(11-3)の３条件のいずれかを満たす場合、連結可能性パラメータ $Conn$ に正の値を設定する。尚、文献 14)では、例えば(11-1)式において、$E_o(q,n)-E_o(q-1,n)\le L_{dif}$という条件を提案していたが、過剰に連結される傾向が見られるため、本稿では、$|E_o(q,n)-E_o(q-1,n)|\le L_{dif}$と絶対値付きに改める。

$|S(P(q),n)-S(P(q-1),n)|<N_{dif}$かつ $E_o(q-1,n)\ge L_{min}$ かつ $E_o(q,n)\ge L_{min}$
かつ$|E_o(q,n)-E_o(q-1,n)|\le L_{dif}$を満たす場合、$Conn$=1　　(11-1)

$|S(P(q),n-1)-S(P(q-1),n)|<N_{dif}$ かつ $E_o(q-1,n)\geq L_{min}$ かつ $E_o(q,n-1)\geq L_{min}$
かつ $|E_o(q,n-1)-E_o(q-1,n)|\leq L_{dif}$ を満たす場合、$Conn$=2　　(11-2)

$|S(P(q),n+1)-S(P(q-1),n)|<N_{dif}$ かつ $E_o(q-1,n)\geq L_{min}$ かつ $E_o(q,n+1)\geq L_{min}$
かつ $|E_o(q,n+1)-E_o(q-1,n)|\leq L_{dif}$ を満たす場合、$Conn$=3　　(11-3)

上記連結条件のしきい値の標準的な設定値は、N_{dif}=6 [単位：１半音を M_o (=25)とする微分音]、L_{min}=1［単位：ベロシティ］、L_{dif}=10［単位：ベロシティ］である。

$Conn$>0 の場合、続いて、既に連結処理が進行している先頭の可変解析フレームを q_o 番目とし、これに対して上記 q 番目の解析音素の連結可能性を判断する。q_o 番目の可変解析フレームにより周波数解析されたノートナンバーn の解析音素の成分を［時刻 $Time(q_o)$, 時間長 $Length(q_o)$, 主周波数 n, 副周波数 $S(P(q_o),n)$, 強度 $E_o(q_o,n)$］とする。q_o 番目の解析音素と q 番目の解析音素との時間的なギャップ $Time(q)-(Time(q_o)+Length(q_o))$が T_{gap} 未満で、ノートナンバーn において上下±1 の変移を考慮し、副周波数を考慮した、q_o 番目と q 番目の可変解析フレームとの副周波数の差が所定値 N_{adif} 未満で、両者の連続性が認められる場合、即ち、以下(12-1)～(12-3)の３条件のいずれかを満たす場合、q 番目の解析音素を q_o 番目の解析音素に連結する処理を実行する。即ち、q_o 番目の解析音素の時間長 $Length(q_o)$を $Time(q)+Length(q)-Time(q_o)$に更新し、$q$ 番目の解析音素を削除する。

$Conn$=1 かつ$|S(P(q_o),n)-S(P(q),n)|<N_{adif}$　　(12-1)

$Conn$=2 かつ$|S(P(q_o),n)-S(P(q),n-1)|<N_{adif}$　　(12-2)

$Conn$=3 かつ$|S(P(q_o),n)-S(P(q),n+1)|<N_{adif}$　　(12-3)

連結後の q_o 番目の解析音素の主周波数・副周波数・強度は、

$Conn$=1 かつ $E_o(q,n)>E_o(q_o,n)$の場合、主周波数 n, 副周波数 $S(P(q),n)$, 強度 $E_o(q,n)$に更新し、

$Conn$=2 かつ $E_o(q,n-1)>E_o(q_o,n)$の場合、主周波数 $n-1$, 副周波数 $S(P(q),n-1)$, 強度 $E_o(q,n-1)$に更新し、

$Conn$=3 かつ $E_o(q,n+1)>E_o(q_o,n)$の場合、主周波数 $n+1$, 副周波数 $S(P(q),n+1)$, 強度 $E_o(q,n+1)$に更新する。

上記連結条件のパラメータ T_{gap} として、文献 14)では、512 サンプル（サンプリング周波数 44.1[kHz]の場合、11[msec]）と固定値を設定していたが、本稿ではノートナンバーごとに可変に設定し、解析フレーム長 $T(n,m)$に比例する値を設定するようにした。例えば、$T_{gap}=4\cdot T(n,m)$とする。これにより、低域部の周波数分解能の向上に伴う、高域部の時間分解能の低下を抑止できる。

また、上記連結条件のパラメータ N_{adif} の標準的な設定値は、N_{adif}=8 [単位：１半音を M_o (=25)とする微分音]である。

３．６．　ノートイベント符号化（ピッチベンド符号化）(F)

前節で述べた時系列の解析音素の連結処理は、不連続性が認められるまで後続する複数の解析音素に対して繰り返し行い、最終的に統合された［時刻 $Time(q_o)$, 時間長 $Length(q_o)$,

主周波数 n，副周波数 $S(P(q_o),n)$，強度 $E_o(q_o,n)$］に対して、２つの MIDI ノートイベントに変換する。時刻(q_o) で、ノートナンバーn のノートオン・イベントを発行し、ベロシティ値（0～127）は $E_o(q_o,n)$の最大値を E_{max} として、$128\cdot\{E_o(q_o,n)/E_{max}\}^{1/4}$ で与える。時刻については、Standard MIDI File では、直前イベントとの相対時刻（デルタタイム）を整数値で与える必要があり、その時刻の単位は任意に定義でき、例えば、1/1536 [sec]の単位に変換して与える。そして、$Time(q_o)+Length(q_o)$の時刻で、演奏中のノートナンバーn に対してノートオフ・イベントを発行する。

より表情豊かな MIDI イベントを作成するためには、前節で行った連結処理を行う前の各解析音素の成分を保存しておき、ピッチベンド・イベント（ノートオン後のピッチを 1/100 半音単位で制御できる）あるいはエクスプレション・イベント（ノートオン後の音量を 128 段階で制御できる）として符号化して、ノートオンおよびノートオフのイベントの間に挿入する。例えば、連結統合された q 番目の解析音素の成分を［時刻 $Time(q)$, 時間長 $Length(q)$, 主周波数 n，副周波数 $S(P(q),n)$，強度 $E_o(q,n)$］に対して、直前に隣接する $q-1$ 番目の解析音素の成分を［時刻 $Time(q-1)$，時間長 $Length(q-1)$，主周波数 n，副周波数 $S(P(q-1),n)$，強度 $E_o(q-1,n)$］とすると、ピッチベンドの値を $4096\cdot\{S(P(q,n)-S(P(q-1),n)\}/M_o+4096$、エクスプレッションの値を $128\cdot[\{E_o(q,n)/E_{max}\}^{1/4}-\{E_o(q-1,n)/E_{max}\}^{1/4}]+127$ と設定して、ノートオン・イベント発行後のデルタタイム $Time(q)-Time(q-1)$の時刻にピッチベンド・イベントおよびエクスプレッション・イベントを発行する。

この時、ノートオン・イベントとピッチベンド・イベントおよびエクスプレッション・イベントとはチャンネル番号で対応付けを行う。MIDI 規格では最大 16 チャンネルまで使用できるが、第 10 チャンネルは通常はパーカッション系の非音階楽器に割り当てられているため、このチャンネルを除く 15 種類のチャンネルのいずれかを各ノートイベント、ピッチベンド・イベントおよびエクスプレッション・イベントに割り当てる。そのため、ピッチベンド・イベントおよびエクスプレッション・イベントを使用する場合、同時に発音できるノートイベントは 15 和音に制限される。

３．７．　和音数の調整(G)

MIDI 符号に変換する段階で、MIDI 音源で処理可能な同時発音数についても考慮する必要がある。時間軸方向に発音期間中（ノートオン状態）のノートイベントの個数を連続的にカウントし、例えば 32 和音（前節のピッチベンドを使用している場合は 15 和音）を超えている箇所が見つかった場合は、強制的に優先度の低いノートイベントを削除する処理を行う。基本的には、同時に発音されている各ノートイベント対のベロシティ値とデュレーション値（ノートオフ時刻－ノートオン時刻）の積（エネルギー値）で優先度を評価し、優先度の低いノートイベントを１対ごと削除する方法をとる。

そうすると、図４-(A)に示されるようにノートイベントが隣接するノートイベントと時間的に重複する場合に、図４-(B)のように音脈上重要なノートイベントも過剰に削除されてしまう。そこで、文献 13)では、図４- (C)(D)に示されるように、ノートイベントを分割して部分的に優先度の低いノートイベントの区間を削除する方法を提案した。ノートオン時のベロシティ値に対してノートオン時刻からの経過時間で補正した補正ベロシティ値を算出し、補正ベロシティ値で優先度を評価し、指定和音数以下になるよう優先度の低いノ

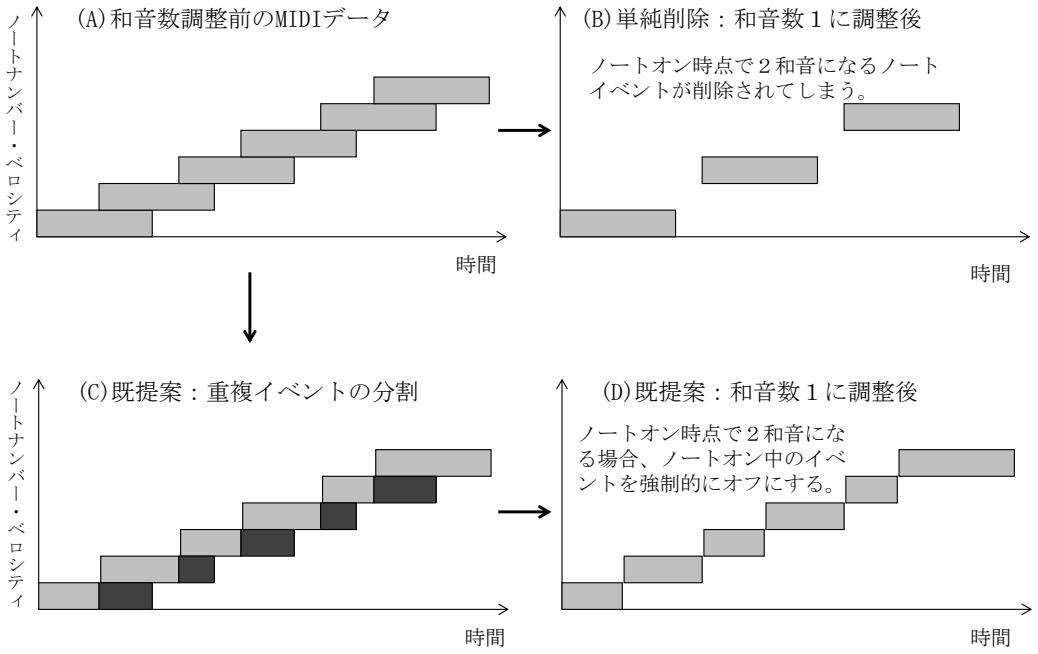


図４　既提案のノートイベントの分割による和音数調整機能

ートイベント対を強制的にノートオフさせる補正処理を行う。この際、ベロシティ値またはデュレーション値のいずれかが所定の下限値より低い場合、優先度に関係無く削除する処理も加える。

i 番目のノートイベント $E_v(i)$のノートオン時刻を $E_v(i).time$、ベロシティ値を $E_v(i).velocity$ とすると、時刻 t ($>E_v(i).time$)におけるノートイベント $E_v(i)$の補正ベロシティ値 $V_c(i,t)$は、

$$V_c(i,t) = E_v(i).velocity \cdot \mathrm{e}^{(t-E_v(i).time)\cdot\tau} \qquad (13)$$

で定義する。τは減衰係数で例えば–1/1536 を与える。（時刻の単位を１秒あたり 1536 とすると、１秒後に 1/2.7 に減衰する。）

３．８．　ビットレートの調整(H)

MIDI データ形式に変換する段階で、MIDI 音源で処理可能なビットレートについても考慮する必要がある。時間軸方向に例えば１秒間隔にノートオンまたはノートオフのイベントの個数をカウントし、各々の符号長を平均５バイト(40bits)とし MIDI 音源で処理可能な最大ビットレートを 9000[bps]とすると（3.6 節のピッチベンドを使用している場合は２倍の 18000[bps]程度に設定する）、１秒間あたりイベント数が 9000/40=225 個を超えている区間が見つかった場合は、その区間に存在するノートオンまたはノートオフのイベントと各々対になるノートオフまたはノートオンのイベントを近傍区間内で探索し、各ノートイベント対のベロシティ値とデュレーション値（ノートオフ時刻－ノートオン時刻）の積（エネルギー値）で優先度を評価し、指定イベント個数（225）になるよう優先度の低いノート

イベント対を局所的に削除する処理を行う。この際、ベロシティ値またはデュレーション値のいずれかが所定の下限値より低い場合、優先度に関係無く削除する処理も加える。

４．評価実験

前章までに述べてきた本稿提案の時間周波数解析法を組み込んだオーディオ-MIDI 変換ツールを、32bits-WindowsAPI (Win32)を用いて 32bits-Windows10/11 デスクトップ・アプリケーションとして C 言語で実装した。本章では正弦波の半音階スケールなどの評価用音源を用いて、本ツールに実装されている時間周波数解析法の改善効果について評価した。尚、本章で提示する 5 点の評価用音源 WAV ファイル（”F5_SinewaveScale88.wav”など）については、本稿末尾で紹介する著者ホームページより、”TestWav_kiyou.zip”という名称で一式ダウンロード可能である。

４．１．　正弦波 88 鍵半音階スケールを用いた周波数解析精度の評価

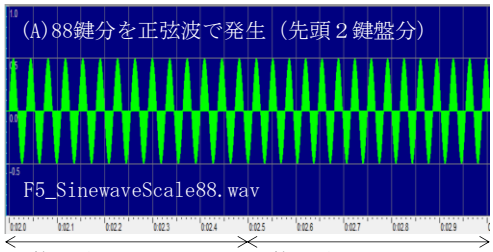
はじめに、正弦波を用いて半音階の解析精度について評価した結果を図５に示す。図５-(A)に示されているように、ピアノの 88 鍵に対応する半音階を 0.5 秒間隔に正弦波で生成した音響波形”F5_SinewaveScale88.wav”（サンプリング周波数 44.1kHz, 量子化 16bits, モノラル）に対し、本稿提案のオーディオ-MIDI 変換ツールを用い、表２の(1)(2)(3)に示される３種類のフレーム長を設定して、MIDI 形式に変換した結果を図５-(B)(C)(D)に示す。

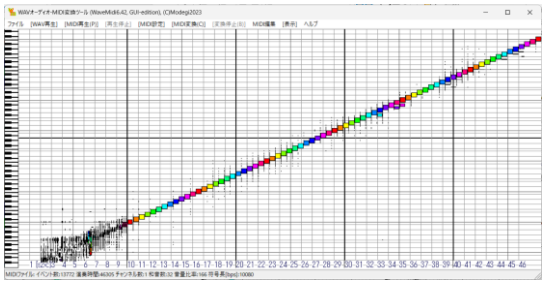
図５-(B)(C)(D)は、変換された MIDI データを独自のピアノロール画面で表示したもので、画面内の着色された小さな矩形が音符（ノートイベント）を示し、横軸は時間で、矩形の横幅はノートオンからノートオフ区間（音価、デュレーション）を示す。縦軸は音高（ノートナンバー）とベロシティを示し、縦方向の矩形の中央位置で音高を示し、縦方向の矩形の高さで強さ（ベロシティ）も示している。矩形の色は半音階の階名に基づいて 12 色に色分けしており、オクターブ違いの音は同一の色になる。

表２で設定されている３種類のフレーム長は、いずれも最低音のノートナンバー21（A-1）を解析できるように設定したものであるが、図５-(B)のフレーム長 2048 の設定では、ノートナンバー36 (C1)未満の音は安定して解析できていない。図５-(C)(D)の設定では、概ねの全ての音階で適切な解析精度が得られているが、図５-(D)のフレーム長 8192 の設定では、ノートナンバー53 (F2)未満の音は、3.5 節で述べた解析音素の連結処理(E)が適切に動作していない。

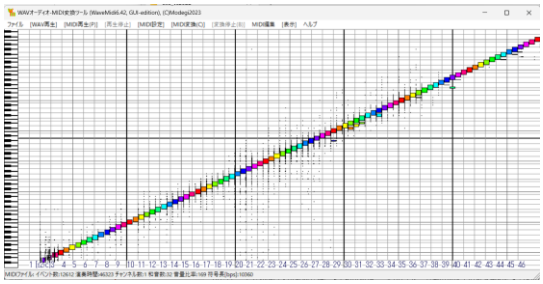
原因は、デフォルトのフレーム長 4096 に対して、フレーム長が 2 倍に設定されているのに、連結間隔のパラメータ T_{gap} がデフォルトの 512 のままに設定されているためである。そこで、図５-(D)に対して、連結間隔のパラメータ T_{gap} をフレーム長に比例して、T_{gap}=512 から T_{gap}=1024 のように 2 倍に設定して MIDI 変換をやり直した結果が図５-(E)で、全ての音階で連結処理が適切に動作するようになっている。ただ、連結間隔のパラメータ T_{gap} を大きくすると、高域の解析音素で連結過剰になり、時間分解能が低下してしまう。そのため、3.5 節で述べたように、連結間隔のパラメータ T_{gap} を文献 14)のように固定値にせず、本稿では $T_{gap}=4\cdot T(n,m)$のように、解析フレーム長 $T(n,m)$に比例する可変な値を設定するようにした。

図５-(E)に対して、連結間隔のパラメータ T_{gap} を $T_{gap}=4\cdot T(n,m)$と可変設定にした結果を、

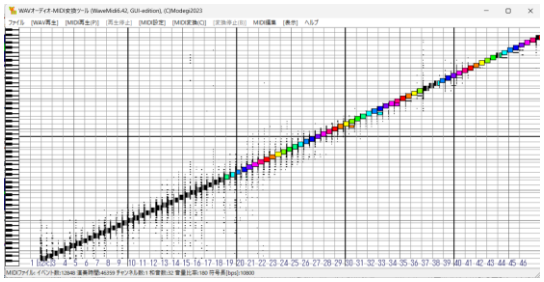




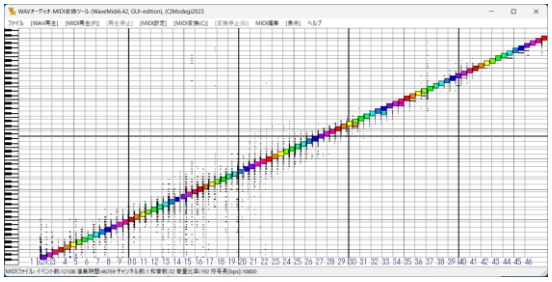
(B)解析フレーム幅2048以下の可変長窓によるMIDI変換
（フレーム長：2048，連結間隔：512）



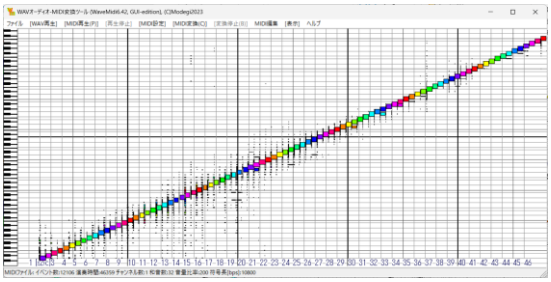
(C)解析フレーム幅4096以下の可変長窓によるMIDI変換
（フレーム長：4096，連結間隔：512）



(D)解析フレーム幅8192以下の可変長窓によるMIDI変換
（解析フレーム幅：8192，連結間隔：512）



(E)解析フレーム幅8192以下の可変長窓によるMIDI変換
（フレーム長：8192，連結間隔：1024）



(F)解析フレーム幅8192以下の可変長窓によるMIDI変換
（フレーム長：8192，連結間隔：可変設定）

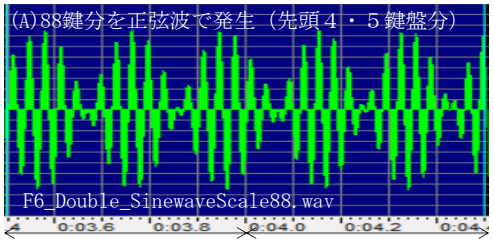
図５　正弦波（ピアノ８８鍵・半音階、0.5 秒×88）の MIDI 変換例

図５-(F)に示す。図５-(F)は、図５-(E)とほぼ同様な結果が得られている。本正弦波の半音階を用いた評価では、時間分解能については評価できないため、連結間隔のパラメータが固定の場合と可変の場合とでは殆ど差が生じない。

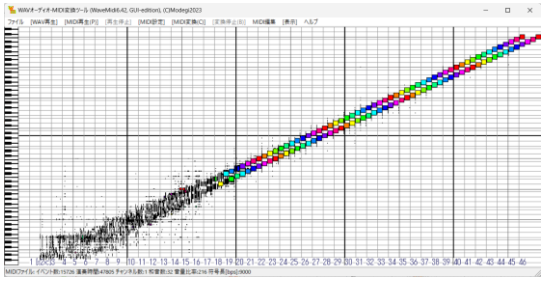
４．２．　二和音半音階スケールと白色雑音を用いた周波数分解能の評価

図５-(C)(D)(E)(F)に示されるように、フレーム長を 4096 以上に設定すれば、概ね 88 鍵の全ての単音は適切に解析できており、3.5 節で述べた、連結間隔パラメータを可変にする効果が得られている。しかし、低域部の周波数分解能については本手法では評価できない。

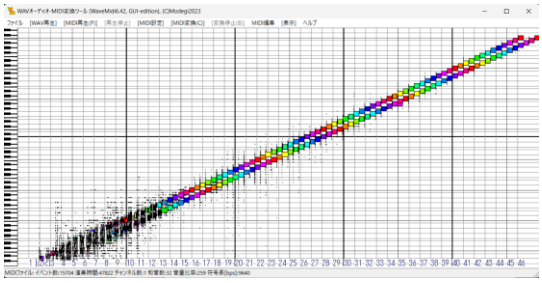
そこで、図６に示すように、正弦波で二和音の半音階スケールを生成し、各和音を構成



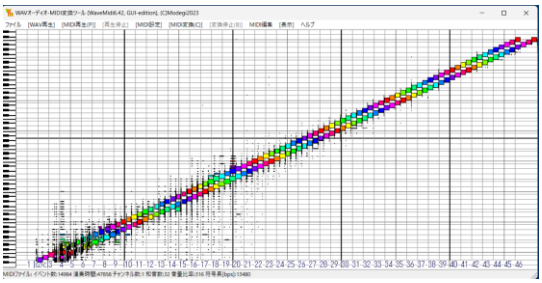
第４鍵 B#-1
（ノートナンバー24）　　第５鍵 C0
（ノートナンバー25）



(B)解析フレーム幅2048以下の可変長窓によるMIDI変換
（フレーム長：2048，連結間隔：可変設定）



(C)解析フレーム幅4096以下の可変長窓によるMIDI変換
（フレーム長：4096，連結間隔：可変設定）



(D)解析フレーム幅8192以下の可変長窓によるMIDI変換
（解析フレーム幅：8192，連結間隔：可変設定）

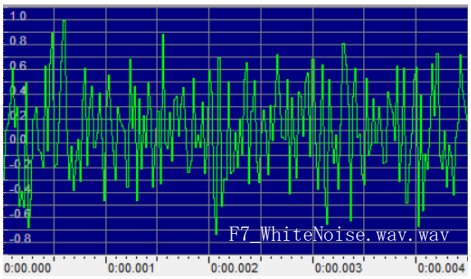
図６　正弦波（ピアノ８８鍵・半音階、0.5 秒×88）二和音の MIDI 変換例

する二音を識別できるか評価を行った。図６-(A)は、図５-(A)の音響信号の２本を互いに 1.5 秒だけずらして合成することにより、音程が 3 半音ずれた短三度の二和音の音階を形成したものである (F6_Double_SinewaveScale88.wav)。そして、図５-(B)(C)(D)と同様に、表２の(1)(2)(3)に示される３種類のフレーム長を設定して、MIDI 形式に変換した結果を図６-(B)(C)(D)に示す。ただし、図６-(B)(C)(D)はいずれも、連結間隔のパラメータ T_{gap} を図５-(F)と同様に可変設定にした。

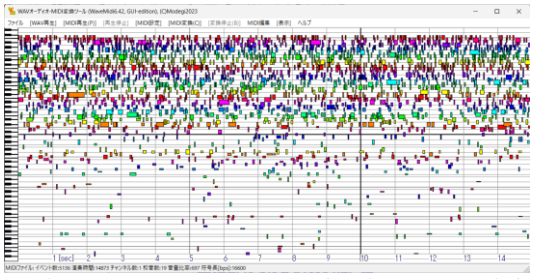
図６-(B)のフレーム長 2048 の設定では、G2/E2（ノートナンバー52 と 55）未満の低音部の二和音は適切に解析できず、図６-(C)のフレーム長 4096 の設定でも、G#1/F1（ノートナンバー41 と 44）未満の低音部の二和音は適切に解析できていない。そこで、図６-(D)のように、フレーム長を 8192 に設定すれば、G0/E0（ノートナンバー28 と 31）以上の低音部の二和音の各構成音も適切に解析できている。即ち、フレーム長は 8192 以上に設定しないと、低域部の 3 半音ずれた二和音の構成音を識別することは難しい。

続いて、和音数を顕著に増やした場合の評価として、白色雑音を用いて解析ムラについて評価を行った。図７-(A)に示す、均一乱数を用いて生成した白色雑音 “F7_WhiteNoise.wav” に対して、図５-(B)(C)(D)と同様に、表２の(1)(2)(3)に示される３種類のフレーム長を設定して、MIDI 形式に変換した結果を図７-(B)(C)(D)に示す。

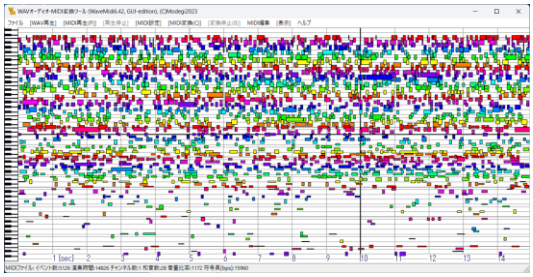
図７-(B)のフレーム長 2048 の設定では、A3（ノートナンバー69）未満の低音部の音は殆ど解析できず、図７-(C)のフレーム長 4096 の設定でも、B1（ノートナンバー47）未満の低音部の音は殆ど解析できていない。そこで、図７-(D)のように、フレーム長を 8192 に設定



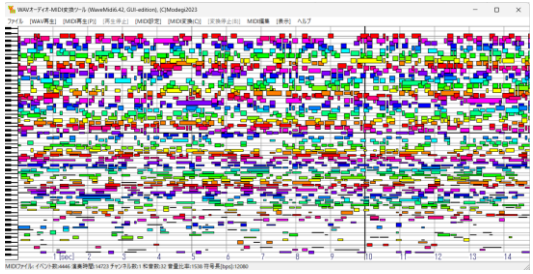
(A) ソース白色雑音：先頭0.0045sec区間を拡大表示



(B) 解析フレーム幅2048以下の可変長窓によるMIDI変換
（フレーム長：2048，連結間隔：可変設定）



(C) 解析フレーム幅4096以下の可変長窓によるMIDI変換
（フレーム長：4096，連結間隔：可変設定）



(D) 解析フレーム幅8192以下の可変長窓によるMIDI変換
（解析フレーム幅：8192，連結間隔：可変設定）

図７　白色雑音の MIDI 変換例 (14.8 sec, F7_WhiteNoise.wav)

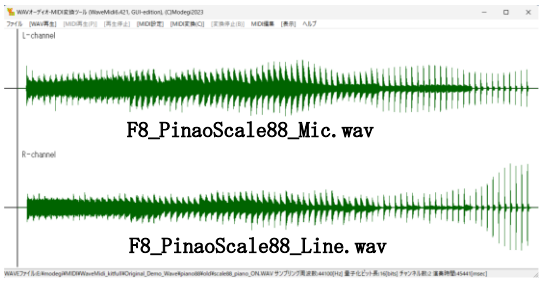
すれば、B0（ノートナンバー35）以上の低音部の音を含め均一に解析できている。従って、白色雑音の場合でも、フレーム長は 8192 以上に設定しないと、低域部の音を含め均一に解析することは難しい。

４．３．　ピアノ音源 88 鍵半音階スケールを用いた周波数解析精度の評価

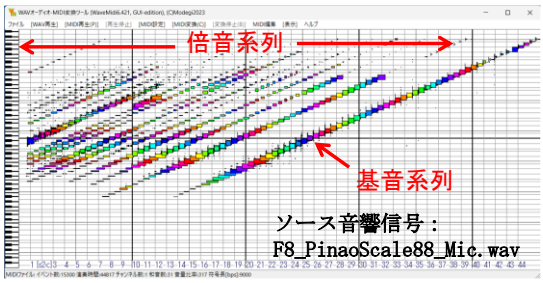
図５と図６の評価においては、ソース音源として正弦波を使用していたが、採譜用途などでは楽器音が使用されることになる。楽器音では、基音だけでなく倍音が加わり、3.3 節で述べたように基音が欠落することもあり、複雑な様相を呈することがある。

そこで、図５-(A)の正弦波の代わりに MIDI 音源のピアノ音を用いて半音階の音響信号を収録して、評価した結果を図８に示す。図８-(A)に示される２チャンネルのソース音響信号は、ピアノの 88 鍵に対応する半音階を 0.5 秒間隔に、Windows に標準搭載されているソフトウェアシンセサイザ（Microsoft GS Wavetable Synth）のグランドピアノ音色（GM No.1）を用いて発音させ、２通りの条件で収録した音響波形（サンプリング周波数 44.1kHz，量子化 16bits, 2 チャンネル）である。

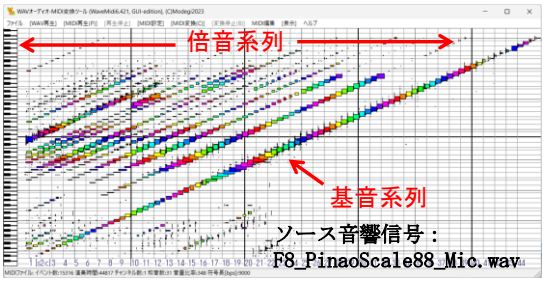
L チャンネル側（F8_PianoScale88_Mic.wav）は、パソコン（Toshiba/DynabookT75）に標準実装されているオーディオデバイス（Realtek High Definition Audio）を用いてシンセサイザ音を再生し、録音を同オーディオデバイス（Realtek High Definition Audio）のマイク配列に設定して、再生音を内臓マイクロフォンで録音したものである。これに対して、R チャ



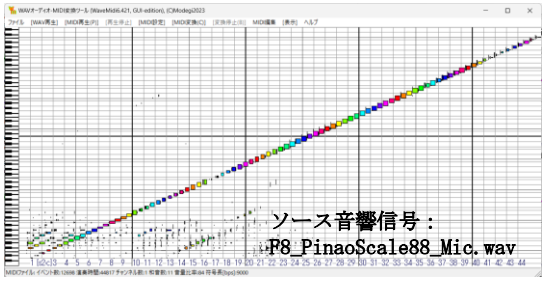
(A)88鍵分のピアノ音を生成・収録（L:Mic，R:Line）
（再生・録音にRealtek High Definition Audio使用）



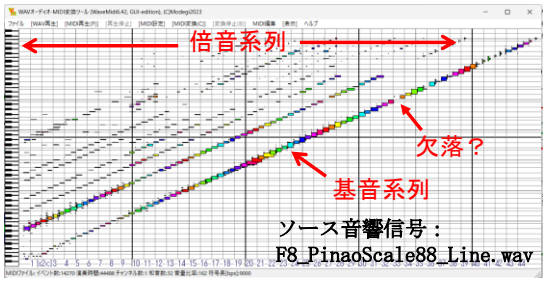
(B)解析フレーム幅8192以下の可変長窓によるMIDI変換
（フレーム長：8192，連結間隔：可変設定）



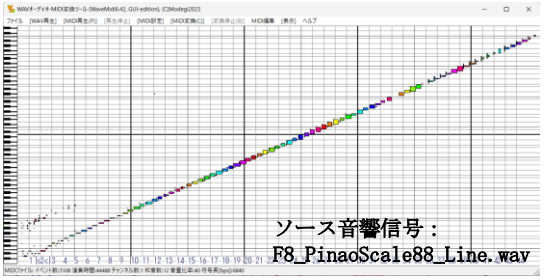
(C)上記(B)に基音推定(最大8倍音)を加えてMIDI変換
（解析フレーム幅：8192，連結間隔：可変設定）



(D)左記(C)に倍音除去(1200%)を加えてMIDI変換
（解析フレーム幅：8192，連結間隔：可変設定）



(E)解析フレーム幅8192以下の可変長窓によるMIDI変換
（解析フレーム幅：8192，連結間隔：可変設定）



(F)左記(E)に倍音除去(1200%)を加えてMIDI変換
（解析フレーム幅：8192，連結間隔：可変設定）

図８　ピアノ音（ピアノ８８鍵・半音階、0.5 秒×88）の MIDI 変換例

ンネル側（F8_PianoScale88_Line.wav）は、同パソコン（Toshiba/DynabookT75）に USB オーディオデバイス（Sound Blaster E1）を外付けし、USB オーディオを用いてシンセサイザ音を再生し、録音を同 USB オーディオの再生リダイレクトに設定し、再生音をそのままライン入力して録音したものである。

図８-(A)の L チャンネルのモノラル信号（F8_PianoScale88_Mic.wav）に対し、本稿提案のオーディオ-MIDI 変換ツールにより、フレーム長 8192 に設定して MIDI 形式に変換した結果を図８-(B)に示す。倍音が含まれているため、右上斜め方向に並ぶ音の系列が図５のように１本にならず、複数本並ぶ形態になり、図示の通り右端に位置する系列が基音になる。図８-(B)の基音系列では B1（ノートナンバー47）未満の音が検出されておらず、これらの基音成分は図８-(A)の L チャンネル信号にも収音されていないことが判明した。即ち、

実装されているマイクロフォンの周波数特性から 123Hz 未満の音は収録できていない。

これらの欠落した基音が、3.3 節で述べたミッシング・ファンダメンタルである。そこで、本稿で追加した基音推定・追加のオプション処理を加え、最大 8 倍音までを推定して、基音を追加補正した結果が、図８-(C)である。図８-(C)では、2 オクターブ下の B-1（ノートナンバー23）までの基音が検出できている。更に、図８-(C)に対して 3.4 節で述べた倍音成分補正を 1200%と大き目に設定して実行した結果が、図８-(D)である。図８-(D)では、図８-(C)の左上に残留していた倍音成分が殆ど削除できている。

これに対して、図８-(E)は、図８-(A)の R チャンネルのモノラル信号（F8_PianoScale88_Line.wav）に対して、図８-(B)と同様にフレーム長 8192 に設定して MIDI 形式に変換した結果である。

図８-(E)では、同様に倍音が含まれているため、右上斜め方向に並ぶ音の系列が図５のように１本にならず、複数本並ぶ形態になり、図示の通り右端に位置する系列が基音になる。基音系列で前述の低域の音を含め、半音階のほぼ全ての音が検出できている。図８-(E)では、C5（ノートナンバー84、図中に「欠落？」と指示）が欠落しているように見えるが、原音を含め発音期間が短いだけで、欠落している訳ではない。同様に、図８-(E)に対して 3.4 節で述べた倍音成分補正を 1200%と大き目に設定して実行した結果が、図８-(F)である。図８-(F)では、図８-(E)の左上に残留していた倍音成分が殆ど削除できている。

このように、本録音条件では、図８-(B)(C)(D)のように、3.3 節で述べたミッシング・ファンダメンタルになる基音は無く、全ての基音が収録できている。

５．おわりに

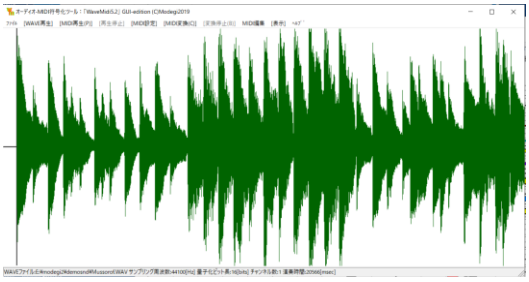
最後に、文献 14)でも使用していた、より実用レベルに近いピアノ独奏音とスピーチ音を用いて、本ツールの改善効果について評価する。

はじめに、ピアノ独奏音の採譜精度について評価した結果を図９に示す。図９-(A)はムソルグスキー「展覧会の絵」ピアノ独奏版について、第１プロムナード冒頭 22 秒のピアノ演奏録音の音響波形（サンプリング周波数 44.1kHz, 量子化 16bits, モノラル）である。

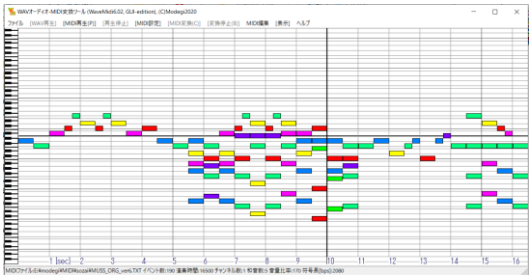
この音源に対応して、採譜の模範解答として、図９-(A)の譜面を基に手作業で MIDI 打ち込みを行った結果を図９-(B)に示す。尚、MIDI 打ち込み時のベロシティ・パラメータは 64 に均一にしている。

図９-(A)に対して、本稿提案のオーディオ-MIDI 変換ツールにより MIDI 形式に変換した結果を図９-(C)(D)に示す。図９-(C)はフレーム長を 4096 サンプルに設定して MIDI 形式に変換したもので、図９-(D)は更に倍音補正 50%を加えた結果である。図９-(B)と比較すると、本音源には 3.3 節で述べたミッシング・ファンダメンタルは存在せず、概ね全ての基音が採譜されている。倍音除去を加えた図９-(D)を図９-(B)と比較すると、高域部の F4（ノートナンバー77）音が欠落しているが、図９-(C)では検出されているので、倍音補正により過剰に除去されたことが要因である。倍音補正の割合を低めに設定すると、上記欠落音以上に残留する倍音が増えてしまうため、倍音補正処理については今後更なる改善が必要である。

図９-(C)(D)では低音部の E1（ノートナンバー40、図中に「連結不良」と指示）音の連結が不安定である。そこで、フレーム長を 8192 サンプルに変更して MIDI 形式に再変換した

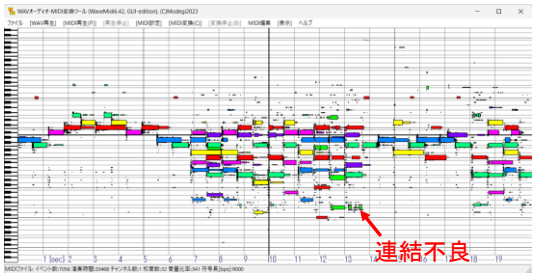


(A)ピアノソロ演奏音の音響波形（冒頭22sec）



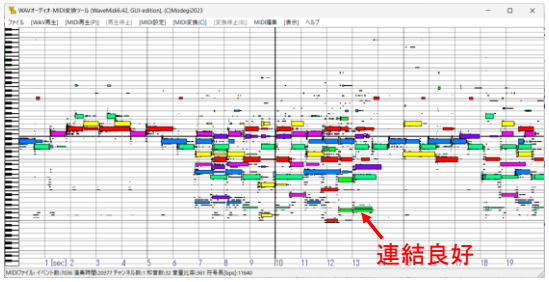
(B)楽曲(A)に対するMIDI打ち込みデータ
（ベロシティ・パラメータ均一）

(C)解析フレーム幅4096以下の可変長窓によるMIDI変換



(D)左記(C)に倍音除去(50%)を加えてMIDI変換

(E)解析フレーム幅8192以下の可変長窓によるMIDI変換

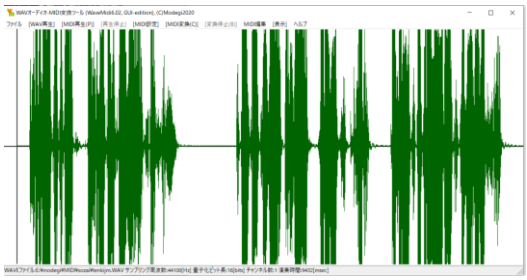


(F)左記(E)に倍音除去(50%)を加えてMIDI変換

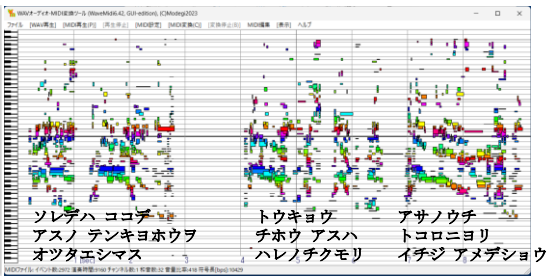
図９　ムソルグスキー「展覧会の絵」ピアノ独奏版の MIDI 変換例

結果を図９-(E)(F)に示す。E1（ノートナンバー40、図中に「連結良好」と指示）音を始め、低音部の音の連結性が改善されている。ただし、図９に示されるテンポが比較的遅い曲（ムソルグスキー組曲「展覧会の絵」ピアノ独奏版、プロムナードなど）では、演奏者のテンポに追従でき、最大６重和音の全てが、演奏者が弾いた通りに概ね再現できている。しかし、本稿では掲載を省略するが、同組曲で第１プロムナードに続く「グノーム（小人）」などテンポが速い楽曲については、フレーム長 8192 では言うまでもなく、フレーム長 4096 に設定しても、倍音除去を行う前段階で正確に拾えない音符があり、周波数解析(B)における時間分解能の更なる改善と解析音素の連結処理(E)の高精度化が今後の課題となる。

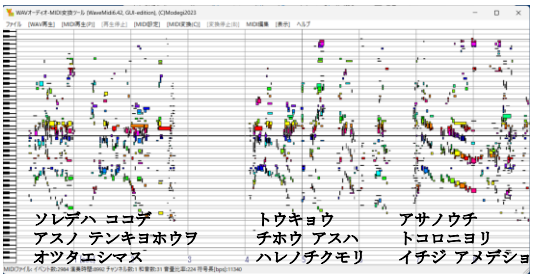
本稿の冒頭で、音楽生成 AI の機械学習向けに、入手が容易な波形オーディオデータを、本ツールにより MIDI データに変換して提供する、生成 AI への応用について言及した。これに加え、AI や機械学習は、本ツールの精度・性能向上にも活用できる。例えば、本ツールにより自動変換された図９-(C)(D)(E)(F)の MIDI データに対して、 図９-(B)を教師データとして機械学習させれば、演奏ゆらぎを自動的に除去して譜面データを生成する自動採



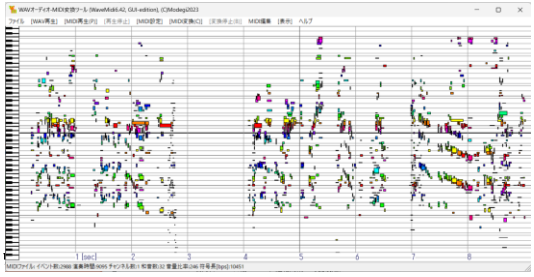
(A)男声アナウンスの音響波形（9.4sec）



(B)解析フレーム幅4096以下の可変長窓によるMIDI変換
（フレーム長：4096，連結間隔：可変設定）



(C)解析フレーム幅2048以下の可変長窓によるMIDI変換
（フレーム長：2048，連結間隔：可変設定）



(D)左記(C)に基音推定(最大2倍音)を加えてMIDI変換

図１０　男声アナウンスの MIDI 変換例

譜機能を実現できる可能性がある。特に、本ツールにおいて前述の課題である、3.4 節の「倍音成分補正(D)」、3.5 節の「解析音素の連結処理(E)」を既定のアルゴリズムだけで精度を向上させるのには限界があり、AI による判断にまかせる方法も考えられるので、今後検討したい。また、ベロシティ情報がない図９-(B)のような単調な MIDI 打ち込みデータに対して、図９-(D)(F)のように、音高や音長に自然な演奏ゆらぎを加えた MIDI 演奏データに自動変換する、打ち込み支援にも活用できる。

最後に、ボーカルの解析精度について評価した結果を図１０に示す。図１０-(A)は男声アナウンス約 9.4 秒の音響波形（サンプリング周波数 44.1kHz, 量子化 16bits, モノラル）である。これに対し、本稿提案のオーディオ-MIDI 変換ツールにより MIDI 形式に変換した結果を図１０-(B)(C)(D)に示し、GM 標準 MIDI 音源（プログラム No.54, “Voice-Ooh”）を用いてボーカル再生音を評価した。図１０-(B)はフレーム長を 4096 サンプルに設定して MIDI 形式に変換したものである。時間分解能が不十分で音声再生音に不明瞭な箇所がみられるので、フレーム長を 2048 サンプルに変更して MIDI 形式に再変換した結果を図１０-(C)に示す。音声再生音は結構聴き取れるようになったが、一部の F0 フォルマント成分が欠落して音素の途切れが目立ち不自然な箇所がある。ボーカルの収音では、マイクロフォン・アンプの周波数特性等から低音部の F0 等が拾えない場合がある。

そこで、3.3 節で述べた基音推定・追加のオプション処理を加え、最大 2 倍音まで推定して、基音を追加補正した結果が、図１０-(D)である。音声再生音は結構明瞭になり、音素の途切れも目立たなくなっている。GM 標準 MIDI 音源（プログラム No.54）を用いて聴取す

ると、図１０-(D)は図１０-(B)(C)よりも明瞭度の良い再生音になっていた。このように、3.3 節で述べた基音推定・追加のオプション処理は、楽器音だけでなくボーカルでも有効であることが判明した。

以上、本稿で紹介したツールおよび、その前身のツールは、過去２２年強にわたって筆者が担当してきた本学・情報表現学科の演習授業の教材として使用してきたもので、並行して改良開発も進めてきた。本学の学生や教職員の皆様とのインタラクションが本研究開発の推進に多大な影響を与えたものと考えており、改めて本学関係者の皆様に謝意を示す。また、前述の通り、本稿で紹介した最新版の Windows 版ツールについては、筆者が担当している「クロスオーバー学習」（旧名称：マルチフィールド体験演習）等の演習授業で既に活用しているが、個人・法人を問わず学外の方にも、以下サイトにて C 言語ソースコードを含めて Windows 版ソフトウェア一式を公開しているので、教育・研究・音楽業務・商用・その他にご活用ください。

［オーディオ-MIDI 変換ツールの公開サイト(2023.9 現在、ver.6.421 を公開)］

以下サイトにて、Windows 版ソフトウェア一式 [WaveMidi_kit_230805.zip]（VisualStudio6.0/2022(x86)プロジェクト形式 C 言語ソースコードと実行ファイル）および同ソフトウェアの基本操作マニュアル[WaveMidi_manual_200922.pdf（2023.8.5 更新）]をダウンロードできます。また、本稿第 4 章の評価実験で使用した 5 点の評価用音源 WAV ファイル（F5_SinewaveScale88.wav など）一式[TestWav_kiyou.zip]と、その他のデモ音源一式[WaveMidi_Demo_230804.zip]についても同サイトよりダウンロードできます。

筆者のホームページ：http://www.bekkoame.ne.jp/~modegi/
または　https://sites.google.com/view/hptoshiomodegi
筆者の連絡先：modegi@bekkoame.ne.jp

引用文献

1) 茂出木敏雄「聴覚芸術への情報学的アプローチと音楽情報処理ツールの開発事例」『尚美学園大学・芸術情報研究』, Vol.18, Nov. 2010, pp.15-35. (https://shobi-u.repo.nii.ac.jp/records/427)

2) 茂出木敏雄「オーディオ-MIDI 符号化ツール「オート符」における表情付け解析機能の実装」『尚美学園大学・芸術情報研究』, Vol.20, Nov. 2011, pp.17-34. (https://shobi-u.repo.nii.ac.jp/records/436)

3) 茂出木敏雄「音響信号の MIDI 符号化ツール「オート符」の Windows10 対応に伴う改修」『尚美学園大学紀要「芸術情報研究」』, Vol.26, Mar.2017, pp.85-104. (https://shobi-u.repo.nii.ac.jp/records/594)

4) 茂出木敏雄「音響情報の MIDI 符号化ツール「オート符」の開発」『芸術科学会誌 DiVA』, No.2, 夏目書房（株）, December 2001, pp.42-48.　（ソフトウェアは一般財団法人デジタルコンテンツ協会（http://www.dcaj.or.jp）より 2010 年頃まで配布、現在は中止）。

5)「採譜の達人」arakisoftware, http://www.pluto.dti.ne.jp/~araki/soft/st.html (2023 年 9 月アクセス).
6) WaveTone (Softonic Developer Hub),「音楽をピアノロール楽譜に起こしてくれる！耳コピー支援ツール」, https://wavetone.softonic.jp/ (2023 年 9 月アクセス).
7) OpenAI/MuseNet, April 25, 2019, https://openai.com/research/musenet (2023 年 9 月アクセス).
8) Google/MusicLM, https://google-research.github.io/seanet/musiclm/examples/ (2023 年 9 月アクセス).
9) 古井貞熙『音響・音声工学』、電子・情報工学入門シリーズ 2、近代科学社、初版第 3 刷、Mar.1996, pp.113-141.
10) 柏野牧夫『音のイリュージョン －知覚を生み出す脳の戦略－』、岩波科学ライブラリー168、岩波書店、初版第 1 刷、Apr.2010, pp.72-92.
11) Toshio Modegi, "Multi-track MIDI Encoding Algorithm Based on GHA for Synthesizing Vocal Sounds," *Journal of Acoustic Society of Japa*n, Vol.20, No.4, April 1999, pp.319-324. (DOI: https://doi.org/10.1250/ast.20.319)
12) 茂出木敏雄「音響信号の平均律音階に基づく汎用解析ツール「オート符」の開発」『電気学会・電子情報システム部門誌』, Vol.123-C, No.10, October 2003, pp.1768-1775. (DOI: https://doi.org/10.1541/ieejeiss.123.1768)
13) 茂出木敏雄「MIDI 符号化ツール「オート符」を用いた音素 MIDI コードの設計と楽器音による音声合成機能の実現」『電気学会・電子情報システム部門誌』, Vol.130-C, No.7, July 2010, pp.1159-1167. (DOI: https://doi.org/10.1541/ieejeiss.130.1159)
14) 茂出木敏雄「時間周波数解析法の精度改善とオーディオ-MIDI 変換ツール開発への応用」『尚美学園大学紀要「芸術情報研究」』, Vol.33, January.2021, pp.51-70. (https://shobi-u.repo.nii.ac.jp/records/721)